MPEG RVC Compliant Intra Prediction for AVC

Karim Maarouf and Ihab Amer
Laboratory of Microelectronic Systems (GR-LSM), EPFL
CH-1015 Lausanne, Switzerland
{karim.maarouf, ihab.amer}@epfl.ch

Abstract

Reconfigurable Video Coding (RVC) aims to provide a framework allowing a dynamic development, implementation and adoption of standardized video coding solutions with features of higher flexibility and re-usability. RVC-CAL is a dataflow oriented language which is used for describing modules in the RVC framework. In this paper, an RVC-CAL description of an AVC intra prediction, module is presented. This module is integrated with inter prediction and entropy coding modules to form an AVC baseline encoder. Development using RVC-CAL is found to be more productive than other traditional approaches. This is due to the abstract and encapsulated representation that is made feasible by RVC-CAL. The module presented is part of an RVC encoding tools library that will be used to generate a BSDL and FND to test the conformance of RVC decoders. It is more convenient to use components from this library than to use reference software or other implementations due to the advantages of RVC-CAL.

1. Introduction

The main aim of the MPEG RVC standard is "to offer a more flexible use and faster path to innovation of MPEG standards in a way that is competitive in the current dynamic environment" [1]. This is meant to give MPEG standards an edge over its market competitors by substantially reducing the time to market. The RVC initiative exploits the reuse of obvious commonalities among different MPEG standards and their possible extensions using appropriate higher level formalisms. Thus the objective of the RVC standard is to describe current and future codecs in a way that makes such commonalities explicit, reducing the implementation burden for device vendors [2]. In order to achieve this objective, RVC suggests simplifying the specification of new coding tools by reusing components of previous standards instead of defining new ones. This is achieved by using a dataflow oriented language, namely RVC-CAL for component description.

In this paper, an RVC-CAL description of an AVC intra prediction module is presented. This module is integrated with inter prediction and entropy coding modules to form an AVC baseline encoder. The modules are part of an encoding tools library that will be used to form encoders that generate RVC complaint bit-streams.

The module presented, along with the integrated encoder, can be used to demonstrate that the existence of RVC encoding tools supports the evolution of the RVC standard. Many benefits can be achieved by supporting the RVC framework with such encoding tools [3]. This includes:

- Instead of modifying the available C/C++ software reference model of a specific MPEG standard to make it able to generate the BSDL bit-stream and the FND decoder description in order to test the conformance of a corresponding RVC decoder, building an RVC encoder using RVC-CAL is more convenient.
- Building the encoder using RVC-CAL enables the usage of the commonalities between many components within various MPEG standards. Hence, the existence of an informative Video Tool Library (VTL) for encoders may be advisable.
- This allows for the construction of "reconfigurable encoders" using the encoder's informative VTL, which specifies the set of Functional Units (FUs) that may be interchangeably combined and connected to form different video encoders, with various compression performances and implementation complexities.

2. Intra prediction

In AVC intra prediction, a predictor P is formed from previously encoded and reconstructed macroblocks. In luma prediction, P can be formed either for every 4x4 block in a macroblock (16 in total) or for the 16x16 macroblock as a whole. There are 9 prediction modes for 4x4 and 4 for 16x16. The encoder selects the mode which minimizes the difference between P and the original block to be encoded [4].

Figure 1 shows how the predictors are calculated from the previously coded samples “A-M” in each of the nine 4x4 modes and figure 2 shows how the entire 16x16 block is predicted in a single operation in each of the four modes.

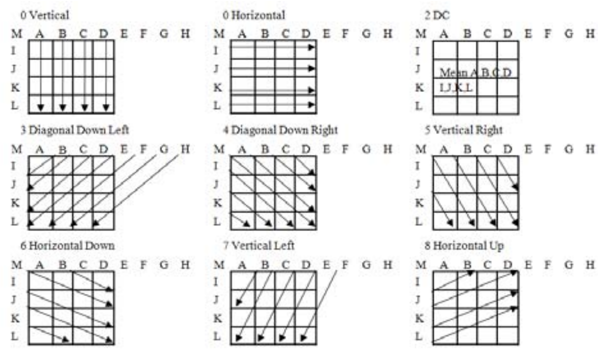

Figure 1 – 4x4 modes

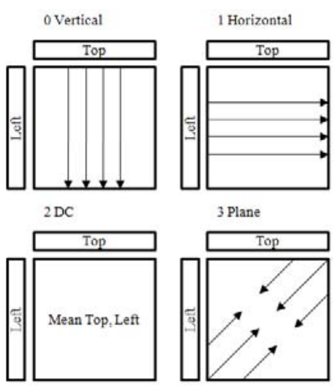

Figure 2 – 16x16 modes

4x4 Modes:

0. A, B, C & D are extrapolated vertically.
1. I, J, K & L are extrapolated horizontally.
2. DC Mode: One value is used for all predictors, which is the mean value of A, B, C, D, I, J, K, L.

3-8. Predictor value is a weighted average of the samples depending on the direction of the arrows following the pattern:

(W+X+Y+Z+RoundValue) >> ShiftValue

16x16 Modes:

0-2. The predictors are generated similar to the 4x4 modes.

3. Plane Mode: A linear function is fitted to the horizontal and vertical samples.

3. Intra prediction RVC-CAL model

3.1. Intra prediction generation

In order to implement an intra prediction module in RVC-CAL which generates predictors for all the modes, a reconfigurable Processing Element (PE) is used based on the architecture in [5]. A PE can be used to generate predictors for a certain mode. However, using reconfigurable PEs, which can be configured to generate predictors for different modes is more efficient than using a PE for each mode. A four-parallel architecture is used, meaning that there are four PEs that generate four predicted pixels in a single prediction iteration. A PE operates in a certain way depending on the prediction mode.

The module consists of a single controller and four identical series of adders, registers and shifters, each representing one of the four PEs. Figure 3 shows the network of actors representing one of the four PEs that correspond to the architecture in [5].

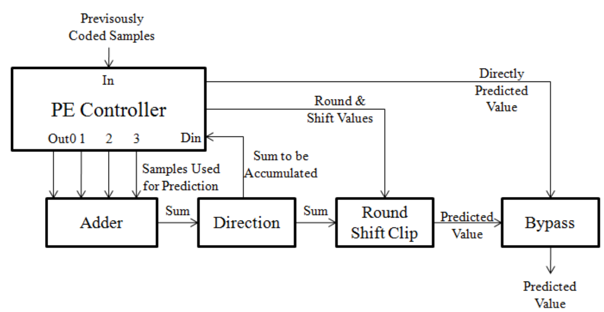

Figure 3 - Intra prediction generation network

The network consists of a *Controller* and an *Adder*, a *Direction* actor, a *Round Shift Clip* actor and a *Bypass* actor. The main actor in the network is the *PE Controller*. It first receives tokens serially at its input port for the previously coded and reconstructed samples, which are used for prediction. For modes requiring no calculations (vertical and horizontal extrapolation), it produces a predictor token directly to the *Bypass* actor. In the DC modes, the predictor value is the mean of the surrounding previously coded samples. The *PE Controller* produces tokens to the *Adder* to calculate the sum of the samples. The sum is calculated over more than one iteration and is accumulated using the *Direction* actor, which produces tokens with the values of the intermediate addition results to the *PE Controller*. The division to produce the mean value is done at the *Round Shift Clip* actor. In the rest of the modes, the predictor is a weighted average of some of the samples. The *PE Controller* produces tokens for the *Adder* and those for the *Round Shift Clip* values depending on the position of the samples being predicted and the prediction mode. At the *Round Shift Clip* actor, the sum from the *Adder* is added to a round value then shifted to get the average and finally clipped to be between 0 and 255.

3.2. Intra prediction mode decision

After generating predictors for the different intra prediction modes for a single MB, a selection of which mode to use must be made. This is known as mode decision. Mode decision is not specified in the AVC standard and is an encoder issue. However, the JM reference software [6] recommends two different methods for mode decision; one of high and one of low complexity. Both methods choose the intra prediction mode using a Lagrangian cost function of both rate and distortion as shown in the following equation.

$$J_{Mode} = Distortion + \lambda Rate$$

The prediction mode chosen is the one that results in the minimum cost (J_{Mode}) for this function. The low complexity mode decision method in the JM reference software is selected, as it is suitable for real time applications. The distortion is calculated using the Sum of Absolute Transformed Difference (SATD) between the original and predicted block. To calculate the SATD, the residue block (R) is first calculated as the difference between the original and predicted blocks. Then, a Hadamard transform is applied to R and the resulting transformed block is summed to calculate the SATD as shown in the following equations.

$$SATD = \left(\sum_{i=0}^{3}\sum_{j=0}^{3}\left|TR_{ij}\right|\right) >> 1$$

$$TR = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{vmatrix} \left|R\right| \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{vmatrix}$$

The rate is calculated as the number of bits required to code the mode decision.

Figure 4 shows the network of actors used to perform the mode decision for both 4x4 and 16x16 modes. This network is made up of a group of networks, each representing a functional unit in the mode decision process.

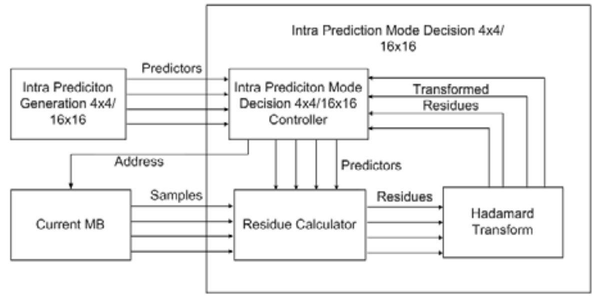

Figure 4 – Mode decision network

As shown in the figure, the *Mode Decision 4x4/16x16 Controller* receives the predictors from the *Intra Prediction Generation 4x4/16x16* network and sends an address to the *Current MB* memory unit, which contains the MB being predicted. The *Residue Calculator* subtracts the predicted samples from the original ones and sends the resulting residues to the *Hadamard Transform* network. The transformed residues are sent back to the Mode *Decision 4x4/16x16 Controller*, which adds them and calculates the SATD.

4. Results and analysis

The intra prediction module was integrated with inter prediction and entropy coding modules to form an AVC baseline encoder. In order to demonstrate the encoding capabilities of the encoder, subjective and objective results of encoding and reconstructing a frame using intra prediction are presented using the Foreman QCIF video sequence.

Figure 5 shows the subjective results of coding the first frame in the Foreman sequence at different QPs. The original frame, the reconstructed frames and the error (difference between original and reconstructed frames) for each QP are shown. The magnitude of the error is scaled by a factor of 10 to make it more visible.

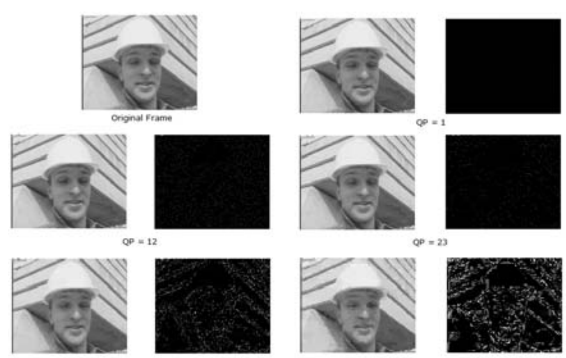

Figure 5 – Subjective results

Figure 6 shows the objective results (PSNR) at the different QPs.

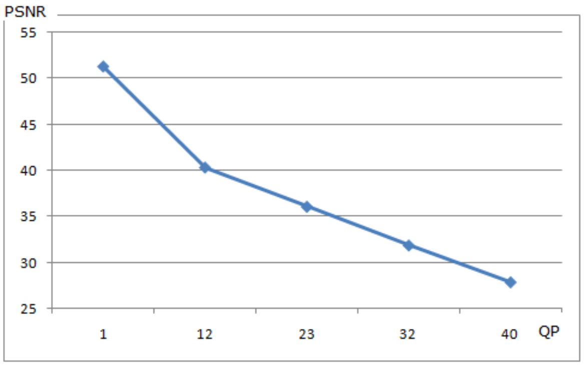

Figure 6 – Objective results

As shown in the figures, as the QP increases (quantization becomes stronger), the quality of the reconstructed frame decreases but the rate decreases.

The intra prediction module can be reconfigured to use only some of the prediction modes. Figure 7 shows the PSNR of the first Foreman frame at QP=10 using a certain number of 4x4 intra prediction modes.

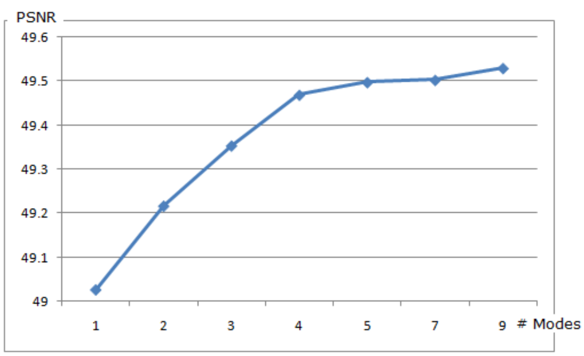

Figure 7 – Objective results using different modes

It is observed that using more prediction modes results in higher PSNRs but needs more computations.

The integrated encoder serves also as evidence of the high productivity and efficiency of using RVC-CAL as a modeling language. When compared to traditional development approaches, RVC-CAL was found to be more productive in terms of development time and the lines of code written. For example, the RVC-CAL description of the intra prediction module has half the lines of code that the corresponding JM reference software implementation has.

The intra prediction module was also synthesized onto a Xilinx Virtex 5 FPGA. After a series of optimization iterations, the resulting implementation was able to encode 30 frames/sec of SDTV (740x480) which is suitable for real-time applications. The synthesis results are shown in table 1.

Table 1 – Synthesis results

CLK Frequency (MHz)	# of Registers	# of LUTs	Throughput (Frames/s)
120.221	5477	6293	30

The results achieved are attributed to a number of factors referring to the strong abstraction and encapsulation properties exhibited by RVC-CAL. Such properties allow developers to focus on one module at a time, without worrying about the order of execution, the synchronization between the different modules or any other process-irrelevant details.

5. Conclusion

In this paper, an RVC-CAL description of an AVC intra prediction module was presented. This module was integrated with inter prediction and entropy coding modules to form an AVC baseline encoder. To demonstrate the advantages of RVC-CAL, the modules were synthesized onto an FPGA. Development using RVC-CAL was also found to be more productive than other traditional approaches. These results were attributed to the abstract and encapsulated representation that was made feasible by RVC-CAL.

The module presented is part of an RVC encoding tools library that will be used to generate an RVC compliant bit-stream to test the conformance of RVC decoders. It is more convenient to use components from this library than to use reference software due to the advantages of using RVC-CAL.

6. References

[1] Jang, E. S., Ohm, J., Mattavelli, M. Whitepaper on Reconfigurable Video Coding (RVC), 2008.
[2] Lucarz, C., Mattavelli, M., Thomas-Kerr, J., Janneck, J. Reconfigurable Media Coding: A New Specification Model for Multimedia Coders, 2007, Workshop on Signal Processing Systems.
[3] Amer, I. Development of RVC Encoding Tools, 2009, MPEG Core Experiment M16409.
[4] Richardson, I. H.264 and MPEG-4 Video Compression (Aberdeen, Wiley, 2003).
[5] Huang, Y., Hsieh, B., Tung-Chien C., Chen, L. Analysis, Fast Algorithm, and VLSI Architecture Design for H.264/AVC Intra Frame Coder. IEEE Transactions on Circuits and systems for Video Technology, 15(3), 2005, 378-401.
[6] Joint Video Team (JVT) reference software, version 14.2, (http://iphome.hhi.de/suehring/tml/download/old jm/jm14.2.zip).